1 RightsDraw

1.1 Introduction

RightsDraw, version 2, is the software prototype developed for providing the proof of concept of a rights management system.

The goal of RightsDraw is to provide a set of services for handling the information on audiovisual rights in various stages of their life-cycle. In particular it provides components for creation, presentation, and editing of rights documents, for making indexes of rights and rights comparison, for formulating queries and for presenting the query results to its users, for managing import and export operations involving the users and or other services.

However RightsDraw does not provide a persistence service for rights information. The rights document are kept on RightsDraw only for the time necessary to run the served activities and then immediately removed on user's request. It is assumed that a persistency service is provided by other systems, such as the PrestoPRIME Preservation Platform (P4) or others, providing equivalent features.

1.2 Document types

1.2.1 Rights OWL documents

The services are operated on MCO/OWL files

The type of serialisation currently supported by RightsDraw is OWL/XML. As explained in [OWL2], it is possible to convert it to/from RDF/XML serialisation and vice-versa, if needed.

Within RightsDraw an MCO/OWL document can represent:

- A PrestoPRIME Rights document – In a single file the complete up-to-date rights situation for an archival item is given, no matter how many contracts contribute to that situation. The Rights holder is implicitly or explicitly the organisation or person holding those contents in the archive. In this document it is expected to find only one main IP-Entity individual plus a number of IP-Entities which are clearly identified as fragments of the main one.
- An MCO contract – In a single file there is a complete contract. The contract can be about multiple IP Entities, which can constitute independent archival items. A single contract may not provide the complete rights situation of an IP Entity
- A Rights Key Pattern – It represents a recurring pattern of rights, which is saved and made available under a label name. These patterns are used by RightsDraw to facilitate the work of its users, see Figure 13 and section 1.6.1.5 . This document only includes deontic clauses without references to Contracts, Users, IP-Entities.
- A Rights Query – A temporary document, similar to a Key Pattern, which is created and used for rights comparison and query. See 1.6.1.3

1.2.2 Mets information packages

RightsDraw can create a Submission Information Package (SIP), as a document in METS format, for supporting the case in which the very first submission to a P4 system is made with only rights information, which are placed in the Rights metadata.

The METS format is also expected for receiving Dissemination Information Packages (DIP) documents from a P4 system, in response to a request posted using an Archival Information Package identifier as key.

1.2.3 Other XML document types

The following XML document types are also in use:

- RightsIndex –used internally to RightsDraw or to a P4 system, as the output of the RightsDraw indexing component
- RightsQueryResults – provided by P4 system to RightsDraw in response to a specific query, addressed by means of the RightsDraw rights comparison component

1.3 Relationships with other services

1.3.1 Relationships with AV Preservation System

In the PrestoPRIME context, the long term preservation services, including persistency of rights information associated to archival items, are provided either by P4 or by an equivalent system.

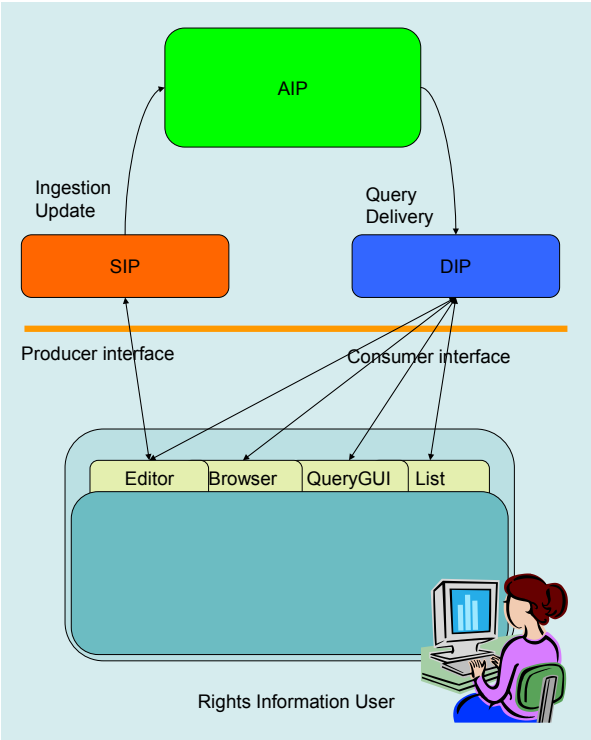

Figure 1 - rights handling and archive

While on one hand RightsDraw serves its human users or other client applications, for rights editing and handling, on the other hand it's a client of P4 services. In order to support rights queries, it is also necessary that P4 has got the RightsDraw components supporting rights indexing and comparison.

Regarding the deployment, the relationship between RightsDraw and P4 is quite flexible, as they can be installed either on independent equipment or on the same host.

The P4 interfaces used by RightsDraw are listed in Table 1.

Interface	Description of use
wf/execute/ingest_rights	SIP submission with only Rights
wf/execute/rights_update	update of OWL for an AIP
wf/execute/query_rights	for posting a rights query
wf/${jobid}/status	for tracking the status of a submitted job
wf/${jobid}/result	for getting the result of a submitted job
search/identifier/${identifier}	for checking about existence of an AIP regarding an IP-Entity
access/dip/${identifier}	for getting a complete DIP
access/dip/${identifier}/rights/result	for getting the URL of the rights document belonging to an AIP
access/dip/list/rights?available=true|false	for having the list of AIPs with/without rights information
access/dip/${identifier}/preview/	for serving a player with browsing quality content

Table 1- P4 interfaces used by RightsDraw

Moreover P4 includes a Rights plug-in, supporting the validation of the OWL document, the creation of the corresponding diagram, the consistency of the Rights section with the Archival Information Package (AIP), and the execution of the Rights indexing (on ingest) and of Rights comparison (on query) stylesheets. Specifically for ingest, P4 provides a workflow implementing the submission of a SIP, containing only Rights.

1.3.2 Relationships with Contract Management System

The PrestoPRIME context doesn't include contracts management. However the creation and the presentation of contracts are activities related to this proof of concept, also because rights information is formulated in MCO.

RightsDraw provides the means for creating and presenting contracts expressed in MCO and its users are expected to examine contracts, in any legacy format, during their activity of rights editing.

By "Contract Management System" we mean here a system in charge for keeping the persistency of the contract documents, in any format, together with the facilities for registering and making access to them.

The relationships between RightsDraw and a Contract Management System are similar to those defined for P4: ingest, update, access, search.

1.4 Architecture

The technical implementation of RightsDraw service is depicted in Figure 2. RightsDraw services are provided to the Users by a normal HTTP server, so that a simple browser can be used, although support of Scalable Vector Graphics (SVG) is required. The scripts deployed under the common gateway interface parse the user requests and run the basic operations consistently. The actual operations on the MCO/OWL files, serialised according to OWL/XML specification, are executed by using the developed XSL libraries.

Some resources, such as the deployed Key Patterns, are shared among the Users, however each User can define his/her settings and owns a part of the document repository.

RightsDraw can also be used by Client Applications.

Figure 3 shows the logical architecture of RightsDraw services, which are organised in various contexts, which are suitable to the various roles covered by the Users.

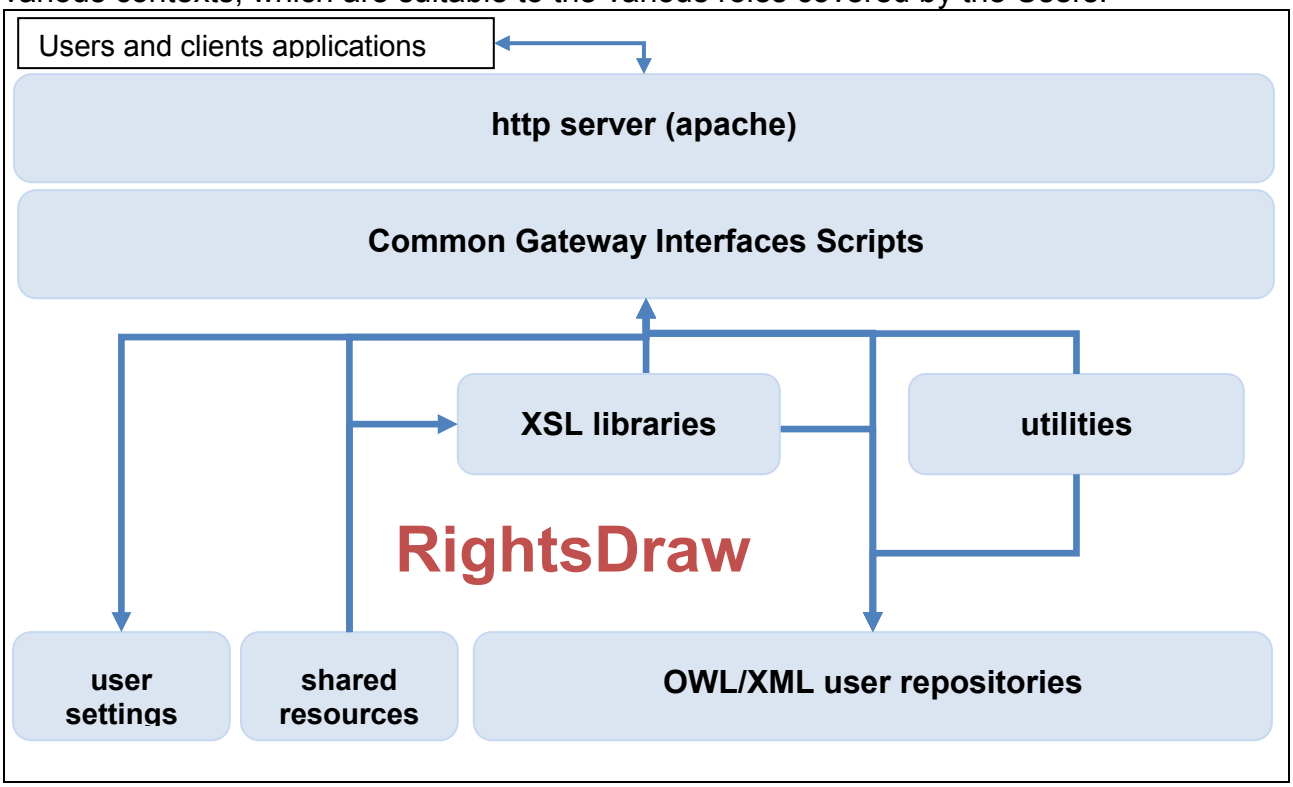

Figure 2 - diagram of RightsDraw architecture

While skilled higher level Users can manage the finest condition details, by working on the diagrams, and can work at the definition and validation of key patterns, other less experienced users can be assigned to simpler rights editing and rights clearance activity, with the support of the deployed Key Patterns, so that their work can be quick and accurate in the same time.

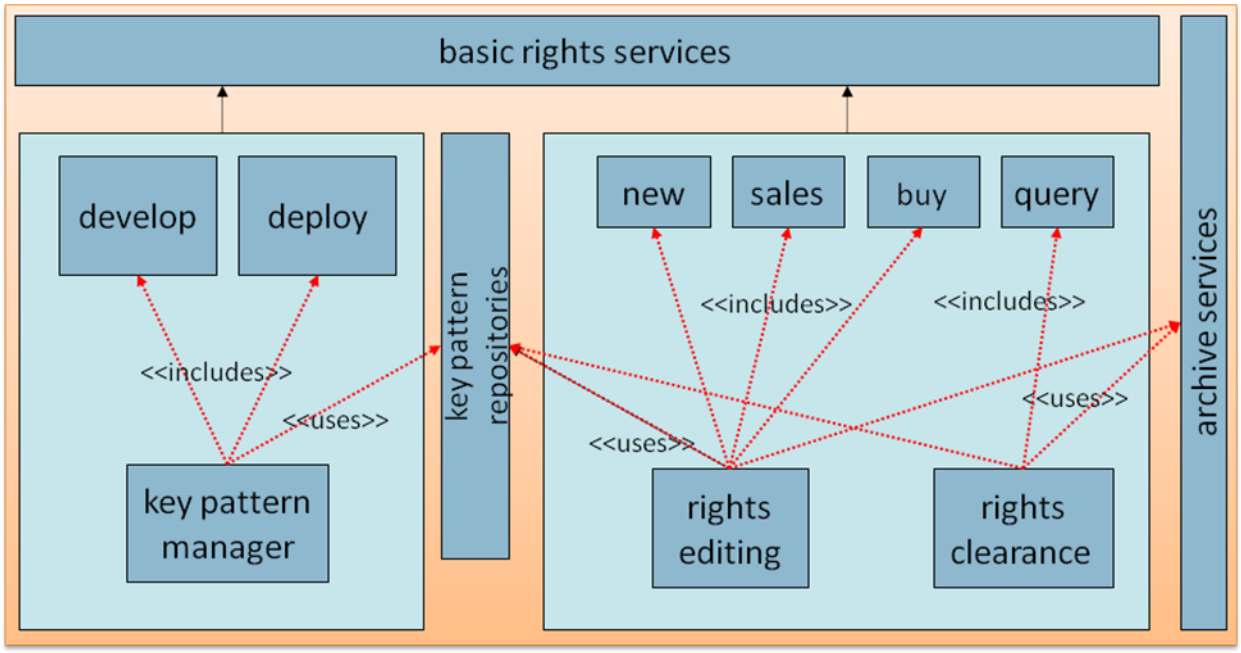

Figure 3 - principles of RightsDraw's use

1.5 Contexts of use

1.5.1 Having Rights information up-to-date associated to archival items

During “Access activities”, the consumer of an audiovisual preservation system, such as P4, can benefit from a huge amount of information. By searching on descriptive metadata, the users can first reach the fruition of the so-called “browsing quality” copy and then they can be granted permission to access the higher levels, technically appropriate to the actions related to the exploitation of an IPEntity, i.e. “communication-to-the-public”, “public performance”, “distribute”, “duplicate”, and “transform” (the action “fixate” has evidently already occurred).

Hence no access to archive is meaningful if it is not possible to establish which are the exploitation rights.

RightsDraw, together with P4, can help to address this issue, by making sure that rights information is actually associated to the archival items, with the following sequence of activities:

- get list of archival items without rights information available
- for each item
  - find all information useful to identify the IPEntity within a contract context
  - find all contracts which are currently relevant to the IPEntity
  - Use Rightsdraw to define completely the rights situation
  - Update the archival information package

1.5.2 Rights clearance

The goal of the activity of Rights clearance is to establish if a target action is permitted, according to the owned rights.

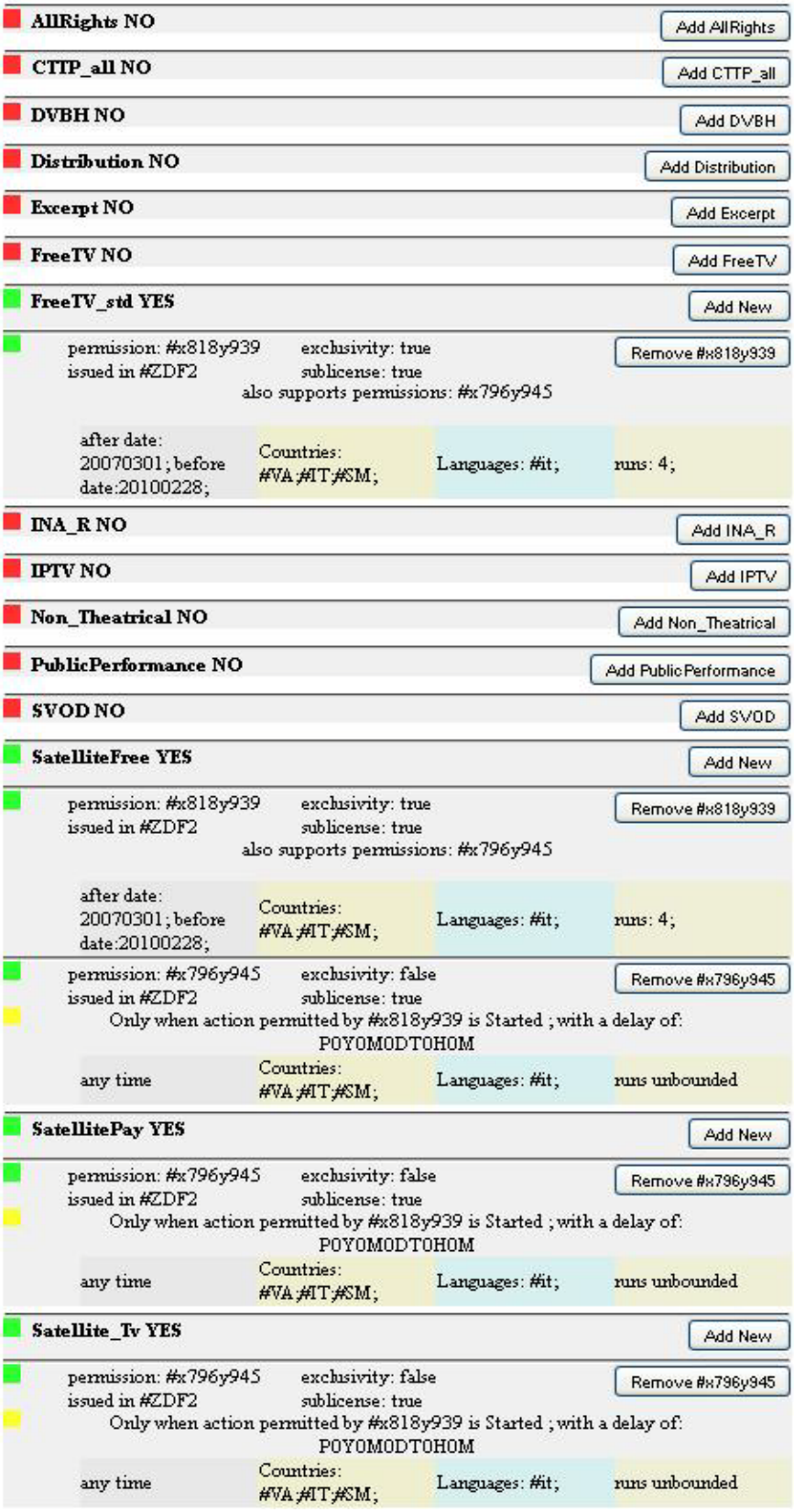

Figure 4 - snapshot of RightsDraw with Key Pattern matching for a sample instance

1.5.2.1 Rights clearance about a specific IPEntity

The simple retrieving of the Rights information set and its presentation in form of graph diagram should be sufficient to fulfil the objective, as the representation of rights is totally unambiguous.

Nevertheless some users might find difficulties in the examination of a diagram, that can become too complex, especially when it needs to reflect real complex rights situation.

The use of the Key Pattern contexts helps the user in this task, as each Key Patterns also represents a clearance scenario. The user can quickly verify what is granted and under which constraints. A snapshot showing this is given in Figure 4.

1.5.2.2 Finding IPEntities matching an exploitation context

The goal of this business case is to find out archival items, the rights of which are appropriate within a target exploitation context. As an example, let's assume that a broadcaster organisation intend to set up a new service, by means of specific technology, e.g. satellite or internet, and need to plan the use of archival content for such service.

This is supported by RightsDraw, together with P4.

The user of RightsDraw defines his/her target exploitation in terms of a query to P4,which is in charge of providing persistency and indexes of the Rights information.

P4 will execute the RightsDraw rights-comparison component and will provide the list of matching items.

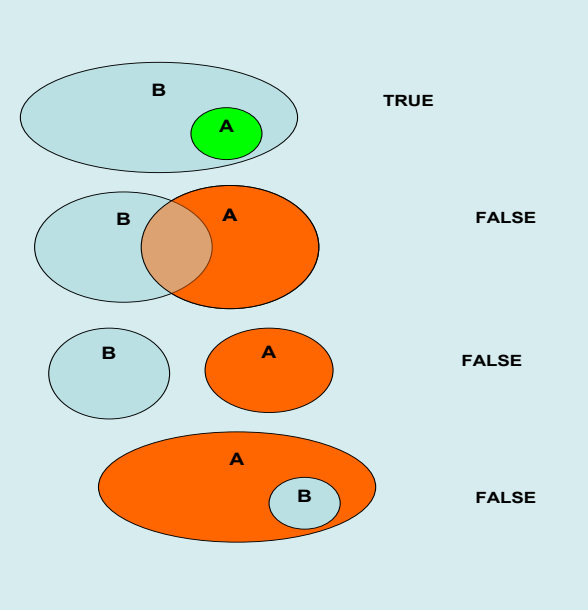

Figure 5 - rights comparison results

Figure 5 shows a representation of the comparison between two Rights situations, as it is required in query matching evaluation. Assuming that **A** is the Rights expected from the query, i.e. the user wants to find all IP-entities having Rights matching **A**, for a given IP-entity with Rights expressed by **B**, it must be possible to provide a result of the probe.
The analysis tells that the result is of boolean type, i.e. either it is True or False, which implies that no ranking values can be given in output.

Actually the result will be True for all cases where **B** is larger or equal to **A**, and False otherwise.

Figure 16 shows how the results of a query on rights are presented, with the link to the video preview, some descriptive metadata and the rights graph, and the buttons for retrieving the selected OWL document.

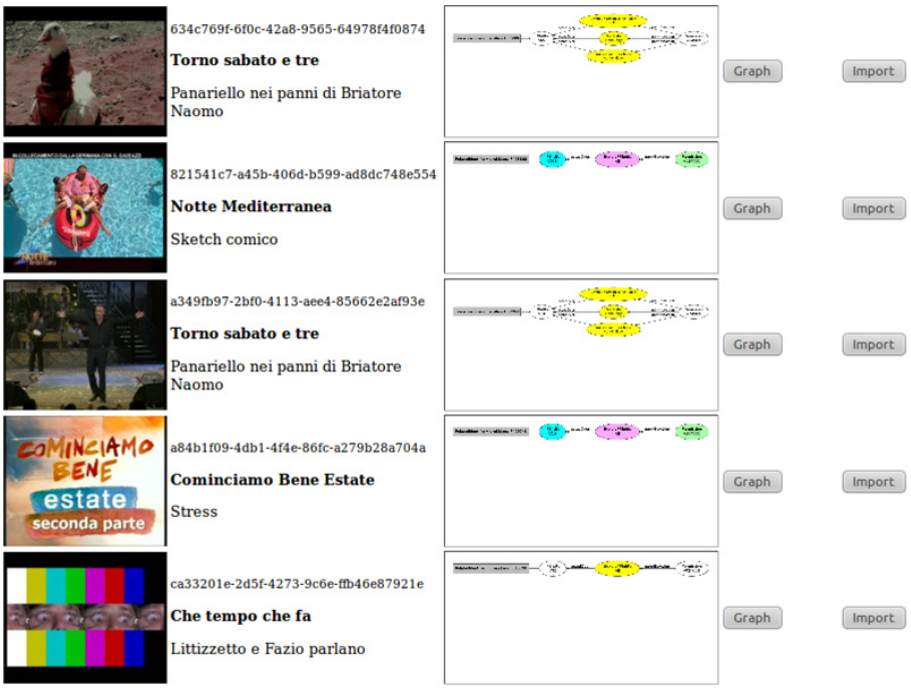

Figure 6 - presentation of results of a query on rights

1.5.3 Dealing with Rights, purchases and sales

Once the Rights situation for an archival item is well represented and saved, the subsequent goal is to keep it up-to-date.

New contracts are the primary reason for making Rights information needing to be modified. As the main objective of MCO is exactly to represent contracts, we can assume that a new contract is expressed according to MCO in OWL.

RightsDraw supports the creation of an MCO contract, however we want to address here the process of applying the deontic expressions defined in a contract to the rights information set associated to an archival item.

1.5.3.1 Purchases

The intention to purchase new Rights might have been the result of a Rights clearance activity or it may rise from other business planning considerations. If we assume that the IPEntity is already known within our organisation, e.g. a copy is held within the archive and some rights are already owned, the practical result of the purchase is that further Permissions have simply to be added to the existing information set.

The fact that the new Permission might have some overlaps with pre-existing ones should not be surprising, as they are issued from distinct contracts, in different times and possibly with different Parties.

1.5.3.2 Sales

Any sale activity must be preceded by a Rights clearance verification not only about the availability of the rights which are intended to be granted, but also about the possibility

to sublicense those Permissions. Indeed it is possible to have a Permission without sublicense right, because the original copyright holder wanted to grant the Permission without exclusivity and keeping only for him/her the permission to trade[1].

A sale will affect the owned rights only if the Permissions are granted with Exclusivity.

In that case, however, to apply a contract is not simple because it may be required to modify a pre-existing Permission, splitting it into two or more ones, for taking into account the resulting conditions.

1.5.4 Migration from legacy systems

The basic problem with most legacy rights management systems is that they have a rights model implemented in a database that could not provide the flexibility required by the evolution of the real rights contract domain and the European legal framework over time.

A quick and effective migration from a legacy system into a new MCO based one is a pre-condition for taking the benefits of the features and functionalities made available.

It is conceivable to design and set up such a process, by developing client applications of RightsDraw. Such applications, however, will need to take into account the characteristics of the legacy systems and the specifications of their export format.

A triage is required in order to identify all the IPEntities for which an automatic migration is possible, without any research and analysis activities on old contracts, that will be run apart. The priority should be given to IPEntities: for which the Rights are not expired; with all rights owned or simpler situations; without ambiguities, which occur when conditions were saved within unstructured textual fields into the legacy systems.

1.6 Manuals

The manuals of RightsDraw are included in the package and can be consulted by the users during their work.

1.6.1 User manual

1.6.1.1 Introduction

RightsDraw is a set of services for creating and handling documents in Media Contract Ontology (MCO), according to ISO/IEC 21000-21 (MPEG-21 part 21).

In the contexts which are supported, a MCO document in RightsDraw can be:

- a Contract, related to the exploitation rights of any set of audiovisual items, with indication of the contract parties, and optionally the inclusion of the narrative contract;
- a rights information set, related to the hold exploitation rights regarding a specific audiovisual item, possibly derived from several contracts.

[1] Exclusivity and Sublicense are modelled as DataProperties, of boolean type, of the Class Permission.

The services are multi-user, i.e. each user owns and manages his/her MCO documents. All users share a set of key patterns rights (KPR), which can be defined in order to make some type of work easier. Only the authorized administrative users can manage the KPR area.

RightsDraw doesn't provide a service of persistent repository, for which it relies on the services of the PrestoPRIME Preservation Platform (P4).

1.6.1.2 MCO services context

From this context the User has the possibility to create, view, and edit a MCO Document with his/her repository area. It is also possible to remove documents and to get out copies of them in export.

The User's repository area is split into two sections, one for contracts and one for pprights (rights related to an archival item). The User can select the working area from the settings section. The document creation form is depending on which working area is in use.

Creating a Contract document

For initiating a new contract document no input is required. A new MCO document, with an automatically generated filename, will be created with an empty contract individual.

By clicking on ellipse representing the contract individual, the User will get forms for adding other contract elements, such as:

- Contract metadata, such as the contract date;
- Contract Parties, either Organizations or Persons;
- Deontic Expressions (Permissions, Obligations, Prohibitions)
- The textual version of the contract (loading a TXT file)

Creating a pprights document

For creating a pprights document, the User has to provide an identifier of the Intellectual Property Entity which is going to be the subject, and the name, of the MCO document. Other information as Title or Description are optional and can be given later.

1.6.1.3 Queries

This context of RightsDraw provides the means to define a rights specific query, post it to the P4 connected system, and examine its results.

It is required that the P4 connected system, which holds the persistent repository, has installed theRightsDraw components for:

- rights indexing
- rights comparing

This functionality only works with MCO pprights documents.

Available query forms

The available rights query forms are the following:

- rights availability - which is used for knowing which archival items from P4 have or not rights information associated;
- query based on KPR - in which the User defines a rights context for a key pattern right, to be used as query to P4;
- free query - in which the User defines a query as a target permission;

Rights comparison

The rights comparison performed on P4 service by the RightsDraw components returns the list of IP entities having at least one permission matching the query target permission.

The comparison between two permissions returns a boolean result:

- true - when the candidate permission is equal or wider (less restrictive) than the target permission
- false - when the candidate permission is narrower (more restrictive) than the target permission

User's actions on query results

The User can first browse the matching items list, including information useful to perform a selection. Any item can then be imported from P4 into RightsDraw for editing.

When the User identifies archival items lacking rights information, he/she can use the information obtained from P4 for initiating a MCO pprights document which, once completed, can be submitted for updating the archival package.

1.6.1.4 In and out

This context of Rightsdraw is for managing the input and output operations regarding MCO documents in the User's repository.

The basic operation with MCO documents are:

- Load - by which the User can upload a document on his/her working area;
- Export - by which the User can get a copy of a document from his/her working area;
- Delete - by which the User can remove a document from his/her working area;

Submission from RightsDraw to P4 interface

The User can save a MCO document from his/her working area in RightsDraw on the preservation system, by submitting it either to the Ingest or to the Update interface of P4.

This functionality is limited to MCO pprights documents.

RightsDraw automatically detects which is the case:

- First submission - a METS Submission Information Package (SIP) is created, with the MCO pprights document embedded in the rights metadata section of Mets. Information about the IP Entity, required to be included in the Mets wrapper, are copied from the MCO document.

- Update - the MCO pprights document is simply posted to the appropriate P4 interface

Import to RightsDraw from P4/Access interface

The User can retrieve the MCO document associated to an archival item preserved on P4 for saving it locally into his/her working area in Rightsdraw for subsequent modifications

The information required by P4 is the Archival Information Package identifier (AIP-ID) that can be easily obtained either from a query or from the P4 GUI.

1.6.1.5 Editing with Key Pattern Rights

This context of Rightsdraw is for creating and editing MCO pprights documents on the basis of the available defined Key Pattern Rights (KPR).

This functionality is limited to MCO pprights documents.

The fact of using this service or subsequent modifications to the definitions of the KPRs does not impact on the usability of the MCO document by other components, and viceversa.

For each defined Key Pattern Right the User can easily verify if his/her MCO document matches it or not, apart the constraints normally associated to the exploitation context, such as Territory, License Period, Language, or Run.

The User can add Permissions defined on the basis of the Key Pattern Rights, specifying the exploitation constraints.

Any Permission can also be removed.

It is important to notice that a single Permission can satisfy more than one Key Pattern Right.

From this context it is also possible for the User to define a Secondary Right, which is a Permission that is restricted to the status of an Action permitted by another Permission.

1.6.1.6 Managing the Key Pattern Repository

This context of Rightsdraw holds the repository of the defined Key Pattern Rights (KPR), which can be admistrated by a specifically authorised User.

The KPR administrator can here:

- verify the content of the deployed KPR
- deploy a new KPR
- withdraw a KPR

Currently two different repositories are provided:

- test
- actual

The User can select which one to use in MKR context or in query context from the settings.

A Key Pattern Rights is a MCO pprights document, without any IP Entity and currently limited to contain a single Permission. It can be easily created and edited within the MCO context.

1.6.1.7 Applying Contracts to PPRights

This context of *Rightsdraw* permits to update the Rights information about an archival item (*pprights*) according to a MCO Contract about the same IPEntity.

General pre-requisites to be verified are:

- The two OWL documents must include the same *IPEntity*
- One of the Parties in the Contract must be the *Rights Holder* of Permissions in the *pprights* documents

In case of sale, that is when the *pprights* rights-holder is act as the *Issuer* in the contract, it must be also verified whether the contract is applicable, i.e. that at least the issued Permissions must be owned, with possibility to sublicence.

In case of sale, the *pprights* have actually to be updated only if the Permissions issued in the Contract are issued with *exclusivity*.

1.6.1.8 User's settings

Each User of Rightsdraw can set/modify the values of his/her setting properties.

It is also possible to set a property to its default value, which is set in the application configuration.

The properties are:

- Related to the Diagram representing the Rights
    - colors - specifies if the diagram must be in color or not
    - size - specifies the size (in inches) of the SVG diagram
    - omitdata - specifies for which DataProperties the data value must be omitted from the diagram, normally because it can be too large.
    - datanamepos - specifies if the name of a DataProperty must be printed on the edge (arc) or within the box
    - rankdir - specifies the direction for drawing the diagram
- Related to how to edit MCO documents
    - mcomode - specifies if the working area and mode are those of contracts or of pprights
    - mykpr - specifies which repository of KPR to use
    - owliriprefix - specifies the prefix of the ontology IRI to be used when creating new MCO documents
- Related to interaction with P4 – for selecting a specific instance of P4
    - p4account - specifies the UserID to be used for logging in with P4 services
    - p4serverurl - specifies the base URL of the P4 services

1.6.2 Administration manual

1.6.2.1 Configuration and installation

Preparation

The *Rightsdraw* software package is distributed in form tarball (tar.gz), and has been tested on a Linux Ubuntu system. The configuration script requires the *tcsh* interpreter.

In principle the service can run on any Linux system, however the resolution of dependencies and the name of some utility commands may be diffent. If you intend to install *Rightsdraw* on a system with a Linux distribution different from Ubuntu, you will need to modify the dependency check part of the configuration and you may need to create aliases to some commands.

You can unpack the tarball in any location of your filesystem, such as `/opt` or `/usr/local`, where the folder *rightsdraw2* will be created and will become the working directory of your *Rightsdraw* services.

The ownership of the working directory and its path hierarchy will be given to the system's user indicated in the initial configuration. Write permission to some folders will also be given to the system's user running apache web server.

Configuration

Before actually installing the software it is required to run the configuration script, which will set a number of properties, the default values of which can be overwritten, by indicating the desired option from the command line.

To run the configuration, execute the following commands from a terminal:

```
$cd /<yourpath>/rightsdraw2
$./configure [OPTIONS]
```

For knowing the configuration options and their default values, just run:

```
$./configure -h
```

The configuration script will first run a dependency check. In case some package will result missing, the script will notify the problem and exit.

On success, the configuration will create the scripts for completing the installation and for the users administration.

Install / Uninstall

The installation command must be given with administration privileges.

```
$cd /<yourpath>/rightsdraw2
$sudo ./install
```

Within the installation, a user named "default" is created. You will be required to set a password for this user and confirm it. The user "default" is used also as a template for the creation of other users. Do not drop it.

To uninstall *Rightsdraw* it is sufficient to run, as root:

```
$cd /<yourpath>/rightsdraw2
$sudo ./uninstall
```

The uninstall script just makes inactive the services to the http server but it doesn't remove the users files and the *Rightsdraw* working directories.

1.7 Service interfaces

All the details on the services interfaces are given in annex C Service Interfaces.

They are organised accordingly to implemented context:

a. MCO context – used for editing contracts and pprights, with the support of graph diagram
b. Query context – for interacting with P4 for posting queries
c. PPRights Repository context - for interacting with P4 and with the User for interchange activities
d. MKR context – for editing rights with the support of the Key Pattern deployed
e. KPR context – for managing the shared Key Pattern repositories
f. Purchase and sales context – for applying new contracts to archival rights
g. Settings context – for viewing and modifying the user's settings

1.8 Source distribution and license

RightsDraw is distributed by RAI under the terms of the GNU Affero General Public License version 3.0[2]. The source is going to be available on http://www.crit.rai.it/EN/attivita/opensource .

[2] http://www.gnu.org/licenses/agpl-3.0.txt